



دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

# اصول و مفاهیم سیستم عامل

ترم بهمن ۹۸

## فصل دوم: مدیریت فرآیند

(زمانبندی)

نستوه طاهری جوان

[nastoooh@aut.ac.ir](mailto:nastoooh@aut.ac.ir)



## فرآیند

❖ تعریف فرآیند:

○ از لحظه ای که یک کار توسط زمانبند انتخاب و وارد گردونهٔ مراحل مختلف می شود تا لحظه ای که اجرای آن به طور کامل خاتمه می یابد، فرآیند گفته می شود.

❖ درج اطلاعات مربوط به فرآیندها در Process Table توسط سیستم عامل

❖ اختصاص یک درایه در Process Table به هر فرآیند با عنوان PCB (Process Control Block)



## PCB

✓ ساختمان داده ای جهت نگهداری اطلاعات مربوط به فرآیندها، شامل:

- شناسه فرآیند
- اولویت فرآیند
- ثبات های پردازنده
- وضعیت فرآیند
- اطلاعات منابع در اختیار فرآیند
- اشاره گرهایی به قسمت های کد، داده و پشته فرآیند
- اطلاعات مربوط به مدیریت حافظه
- و ...



## حالت‌ها (وضعیت‌ها) فرآیند

✓ حالت اول: New

○ فرآیند در حال ایجاد شدن

✓ حالت دوم: Ready

○ فرآیند درون حافظه، در اختیار داشتن همه منابع، منتظر پردازنده

✓ حالت سوم: Running

○ در اختیار داشتن پردازنده و در حال اجرا

✓ حالت چهارم: Waiting (یا Blocked)

○ منتظر وقوع یک رخداد (مثلا اختصاص یک منبع، یا تکمیل I/O) عدم نیاز به پردازنده

✓ حالت پنجم: Terminated

○ اتمام اجرای فرآیند



## تغییر حالات ممکن برای یک فرآیند

✓ از New به Ready

- سیستم آماده دریافت یک فرآیند جدید
- انجام توسط زمانبند کار (Job Scheduler)

✓ از Ready به Running

- اختصاص پردازنده به فرآیند
- انجام توسط زمانبند فرآیند (Process Scheduler)

✓ از Running به Ready

- پس گرفتن پردازنده از فرآیند
  - فرآیند اختیاری پردازنده را رها کند
  - سهمیه فعلی اجرای فرآیند پایان یابد
  - فرآیندی با اولویت بالاتر از راه برسد



## تغییر حالات ممکن برای یک فرآیند

✓ از Running به Waiting

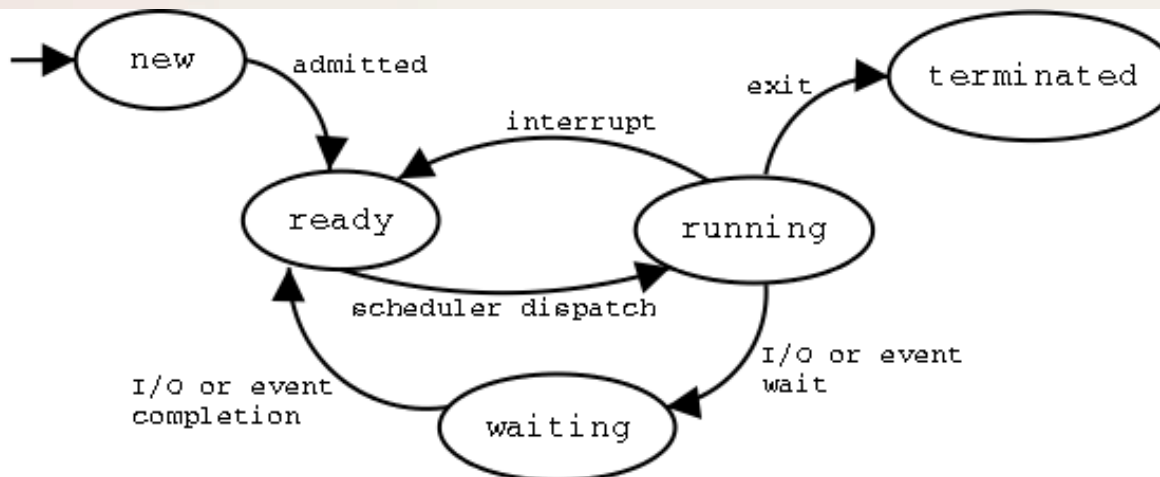
○ انتظار فرآیند برای وقوع یک رخداد

✓ از Ready به Waiting

○ رخ دادن رویدادی که فرآیند منتظر آن است

✓ از Running به Terminated

○ اتمام فرآیند





## وضعیت های فرآیند (تکمیلی)

❖ موقعیت Suspend (معلق) برای فرآیندها:

○ بیرون راندن فرآیند از حافظه به دیسک!

✓ حالت ششم: Suspend Wait

○ فرآیند در حالت انتظار، از حافظه به دیسک منتقل شود

✓ حالت هفتم: Suspend Ready

○ فرآیند به علت کمبود حافظه مستقیم از حالت آماده به آماده و معلق می رود

○ فرآیند بر روی دیسک و در حالت منتظر و معلق قرار دارد، حادثه مورد

نظرش رخ می دهد.



## تغییر حالت های یک فرآیند (تکمیلی)

### از Waiting به Suspend Waiting ✓

- همه فرآیندها در حال انتظارند (هیچ فرآیندی آماده نیست)، یک فرآیند از صف انتظار انتخاب شده و به دیسک منتقل می شود (تا فضا بر روی حافظه ایجاد شده و یک فرآیند از حالت جدید یا آماده و معلق به آماده رود)

### از Suspend Waiting به Suspend Ready ✓

- رخ دادن رویدادی که یک فرآیند منتظر و معلق منتظر آن بود.

### از Suspend Ready به Ready ✓

- هیچ فرآیندی در حالت آماده در سیستم نباشد
- اولویت فرآیند آماده و معلق از همه فرآیندهای آماده بالاتر باشد





## تغییر حالت های یک فرآیند (تکمیلی)

✓ از Ready به Suspend Ready

○ احتمال وقوع پایین

○ فرآیند در حال اجرا به حافظه بیشتری نیاز دارد، از طرفی هیچ فرآیندی در حالت انتظار نیست.

✓ از Suspend Waiting به Waiting

○ خالی شدن حافظه



## زمانبندیها

### ✓ انواع زمانبندیها

- زمانبند بلند مدت (Long Term Scheduler)
  - انتخاب فرآیندها از بین کارها
  - مشخص کردن درجه چند برنامه‌نگی
  - نام‌های دیگر: زمانبند کار (job) یا زمانبند پذیرش (Admission)
- زمانبند میان مدت (Middle Term Scheduler)
  - جابجا کردن فرآیندها از حافظه به دیسک و بالعکس
  - نام دیگر: زمانبند حافظه
- زمانبند کوتاه مدت (Short Term Scheduler)
  - انتخاب از بین فرآیندهای آماده برای اجرا
  - نام‌های دیگر: زمانبند فرآیند یا زمانبند پردازنده



## تعویض متن

### ✓ تعویض متن (Context Switch)

- سوئیچ کردن پردازنده بین فرآیندهای مختلف
- نوعی سرشار، (زیرا از نظر اجرایی کار مفیدی صورت نمی گیرد)
- نیازمند ذخیره کردن حالت فرآیند قبلی و بارگذاری حالت فرآیند فعلی
- انجام توسط ماژولی از سیستم عامل به نام Dispatcher



## زمانبند فرآیند (کوتاه مدت)

### ✓ برخی اهداف

- عدالت
- کاهش زمان پاسخ (Response Time)
  - زمان بین لحظه ورود کار و لحظه شروع پاسخ (اهمیت در سیستم های محاوره ای)
- کاهش زمان گردش کار (Turnaround Time)
  - زمان بین لحظه ورود کار و لحظه خروج کامل
- کاهش زمان انتظار (Waiting Time)
  - مجموع زمان های حالت آماده برای یک فرآیند (انتظار برای پردازنده)
- افزایش بهره وری پردازنده (CPU Utilization, Efficiency)
- افزایش توان گذردهی (Throughput)
  - تعداد فرآیندهای تکمیل شده در واحد زمان



## انواع زمانبندها

✓ غیر قابل پس گیری (یا انحصاری) (Non Preemptive)

- نمی توان پردازنده را به زور از فرآیندی گرفت
- یا باید فرآیند خاتمه یابد، یا فرآیند داوطلبانه پردازنده را رها کند!
- مناسب و کارآ برای سیستم های دسته ای

✓ قابل پس گیری (یا غیر انحصاری) (Preemptive)

- می توان پردازنده را به زور از فرآیندی گرفت
- مناسب برای سیستم های محاوره ای
- اتلاف زمان پردازنده به دلیل تعویض متن های پی در پی



## الگوریتم های زمانبندی فرآیند

✓ تنوع الگوریتم ها:

- MLFQ ○
- LPT ○
- Lottery ○
- Guaranteed ○
- FSS ○
- ... و ○
- FCFS ○
- SJF ○
- RR ○
- SRT ○
- HRRN ○
- Priority ○
- MLQ ○



## برخی تعاریف اولیه

✓ زمان اجرا، زمان سرویس، زمان انفجار (CPU Burst Time)

○ مدت زمان نیاز یک فرآیند به پردازنده

✓ میانگین زمان انتظار

○ میانگین مدت زمان انتظار فرآیندها در صف آماده

✓ میانگین زمان پاسخ!

○ تعریف معمول در تست ها (برخلاف تعریف قبل!) میانگین زمان اجرای فرآیندها، به علاوه زمان انتظار آنها (در واقع معادل زمان گردش کار!)

✓ مشکل گرسنگی، قحطی زدگی (Starvation)

○ احتمال به تعویق افتادن اجرای یک فرآیند تا بی نهایت



## الگوریتم FCFS

- ✓ اجرای کارها با همان ترتیب ورود (مانند یک صف)
- ✓ به نوعی ساده ترین الگوریتم زمانبندی
- ✓ نام دیگر: FIFO
- ✓ از نوع انحصاری (Non Preemptive)
- ✓ نبود مشکل گرسنگی



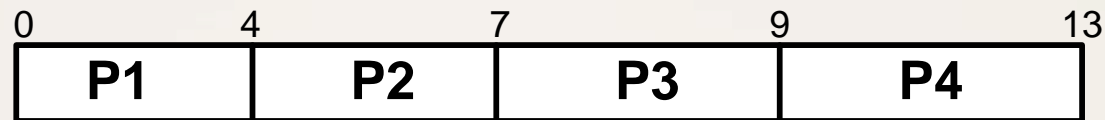


## مثال از FCFS

✓ چهار فرآیند، به ترتیب نامشان همگی در لحظه ۰ وارد سیستم شده اند (با اختلاف اندک) با توجه به جدول زیر، میانگین زمان انتظار برای این چهار فرآیند چقدر است؟

فرآیند	CBT (ms)	زمان ورود	زمان خروج	زمان گردش کار	زمان انتظار
P1	4	0	4	4	0
P2	3	0	7	7	4
P3	2	0	9	9	7
P4	4	0	13	13	9

○ پاسخ کوتاه: نمودار گانت:



$$W.T. = \frac{0 + 4 + 7 + 9}{4} = 5$$



## الگوریتم SJF(Shortest Job First)

- ✓ انتخاب کوتاهترین کار باقیمانده برای اجرا
- ✓ هدف: حداقل کردن میانگین زمان انتظار و زمان پاسخ
- ✓ الگوریتم انحصاری
- ✓ دارای مشکل گرسنگی
  - ورود پیوسته تعدادی کار کوچک و تعویق یک کار بزرگ تا ابد!
- ✓ اعمال ایده FCFS بین دو فرآیند در صورت داشتن CBT یکسان
- ✓ در عمل قابل پیاده سازی نیست
  - زیرا زمان اجرای فرآیندها عملاً از قبل مشخص نیست!
- ✓ نام های دیگر: SPN(Shortest Process Next)
- SPT(Shortest Processing Time)

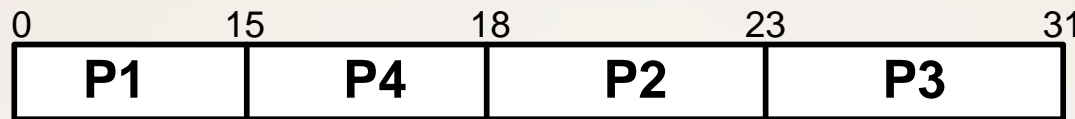


## مثال از SJF

✓ چهار فرآیند زیر را در نظر بگیرید، میانگین زمان انتظار و میانگین زمان پاسخ را برای این فرآیندها با روش SJF به دست آورید.

فرآیند	CBT	زمان ورود
P1	15	0
P2	5	2
P3	8	2
P4	3	2

✓ پاسخ کوتاه:



$$W.T. = \frac{0 + 16 + 21 + 13}{4} = 12.5$$

$$R.T. = \frac{15 + 21 + 29 + 16}{4} = 22.5$$

✓ نکته: زمان پاسخ را (همانند تست ها) همان زمان گردش کار در نظر گرفتیم!



## الگوریتم Round Robin

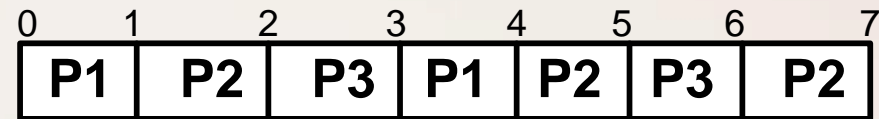
- ✓ تقسیم زمان پردازنده به برش های کوتاه (Time Slice)
- ✓ سرویس دهی پردازنده از ابتدای صف به اندازه یک برش به فرآیندها
- ✓ ورود فرآیندهای تازه وارد به انتهای صف
- ✓ رها کردن فرآیند جاری پس از اتمام برش زمانی
  - انتقال فرآیند رها شده به انتهای صف!
- ✓ قابل استفاده در سیستم های اشتراک زمانی
- ✓ نوع قابل پس گیری FCFS
- ✓ غیر انحصاری
- ✓ نبود مشکل گرسنگی



## مثال از R.R.

✓ سه فرآیند زیر را با استفاده از الگوریتم نوبت چرخشی، و با برش زمانی یک میلی ثانیه زمانبندی کنید. میانگین زمان انتظار و زمان پاسخ فرآیندها را به دست آورید. (فرض کنید فرآیندها در لحظه صفر و به ترتیب نام وارد شده اند.)

فرآیند	CBT	زمان ورود
P1	2	0
P2	3	0
P3	2	0



○ پس داریم:

$$W.T. = \frac{2 + 4 + 4}{3} = 3.3$$

$$R.T. = \frac{4 + 7 + 6}{3} = 5.6$$

فرآیند	زمان پاسخ	زمان انتظار
P1	4	2
P2	7	4
P3	6	4



## نکات حل مسأله R.R.

- ✓ اگر یک فرآیند قدیمی و یک فرآیند تازه وارد، هر دو در یک زمان به انتهای صف آماده برسند، فرآیند تازه وارد جلوتر و فرآیند قدیمی در انتها قرار میگیرد. (در همه سیستم عامل ها رعایت نمی شود)
- ✓ اگر فرآیندی پیش از پایان برش زمانی به پایان برسد، **داوطلبانه** پردازنده را در نیمه برش زمانی رها می کند و پردازنده در اختیار فرآیند بعدی قرار می گیرد.
- ✓ فاصله بین زمان خروج و زمان ورود برابر زمان پاسخ است.
- ✓ اگر زمان اجرا را از زمان پاسخ کم کنیم، زمان انتظار حاصل می شود.
- ✓ **معمولا** زمان تعویض متن نادیده گرفته می شود. (نه لزوما)
- ✓ تغییر اندازه برش زمانی بر روی میانگین زمان پاسخ فرآیندها تاثیر می گذارد.
  - گاهی زمان پاسخ را کاهش می دهد و گاهی افزایش.



## مثالی از R. R.

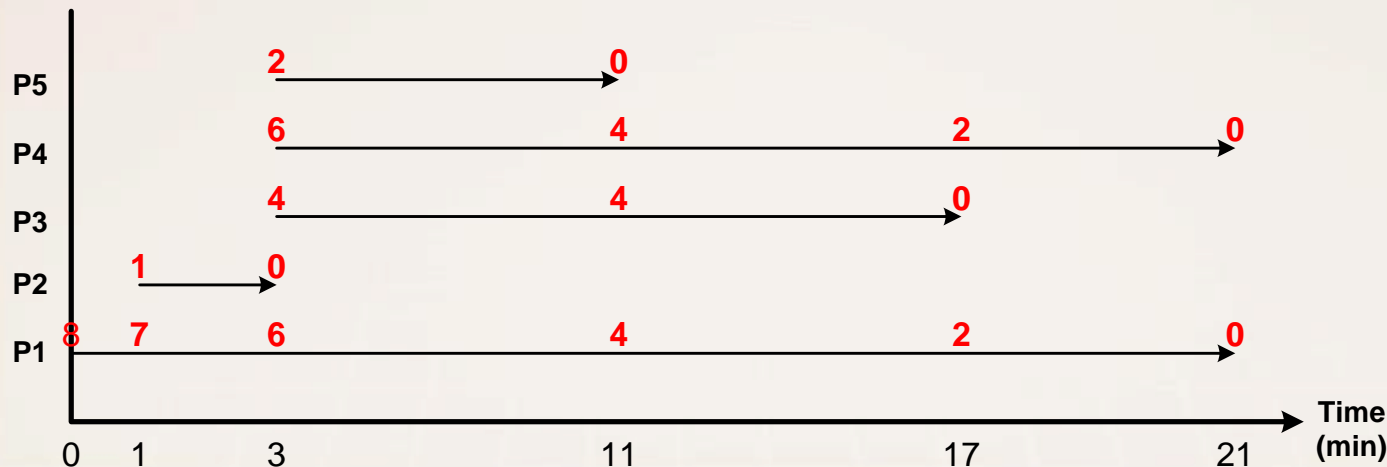
✓ فرآیندهای زیر را در نظر بگیرید. سیستم از روش نوبت چرخشی با برش زمانی یک میلی ثانیه استفاده می کند. میانگین زمان پاسخ را بیابید.

فرآیند	CBT (دقیقه)	زمان ورود (دقیقه)
P1	8	0
P2	1	1
P3	4	3
P4	6	3
P5	2	3

○ نمودار ابداعی.

• اعداد قرمز بالای نمودار: زمان باقی مانده هر فرآیند.

$$R.T. = \frac{21 + 2 + 14 + 18 + 8}{5} = 12.6$$





## الگوریتم SRT

- ✓ نسخه غیر انحصاری (preemptive) الگوریتم SJF
- ✓ پس گرفتن پردازنده از فرآیند در حال اجرا، در صورت ورود یک فرآیند با زمان اجرای کوتاه تر.
- ✓ نام های مصطلح:

- SRT(Shortest Remaining Time)
- SRPT(Shortest Remaining Processing Time)
- SRTF(Shortest Remaining Time First)
- SRTN(Shortest Remaining Time Next)

✓ احتمال وقوع قحطی زدگی برای کارهای بزرگ!!!

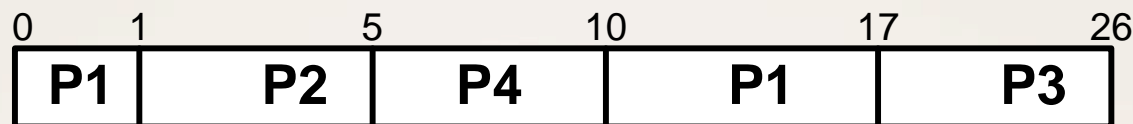




## مثال از SRT

✓ چهار فرآیند زیر را در نظر بگیرید. با روش SRT میانگین زمان انتظار برای فرآیندها چقدر است؟

فرآیند	CBT	زمان ورود
P1	8	0
P2	4	1
P3	9	2
P4	5	3



$$W.T. = \frac{9 + 0 + 15 + 2}{4} = 6.5$$



## الگوریتم HRRN

✓ انگیزه: مشکل قحطی زدگی SJF و SRT برای کارهای بزرگ.

✓ راه کار: دادن اولویت به کارهایی که زیاد منتظر مانده اند.

✓ تعریف اولویت

$$\text{اولویت} = \frac{\text{زمان انتظار} + \text{زمان اجرا}}{\text{زمان اجرا}}$$

✓ در لحظه تصمیم گیری، اولویت ها محاسبه می شوند.

✓ تاثیر پذیری اولویت هم از زمان اجرا و هم از زمان انتظار

○ هر چه فرآیندی کوچکتر باشد، اولوین بالاتری دارد.

○ هر چه فرآیند بیشتر منتظر مانده باشد، اولویت بالاتری دارد.

✓ از نوع انحصاری.

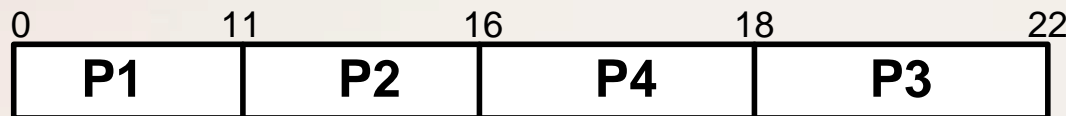
✓ مشکل قحطی زدگی ندارد.



## مثال از HRRN

✓ میانگین زمان پاسخ و زمان انتظار برای چهار فرآیند زیر را به روش HRRN محاسبه کنید.

فرآیند	CBT	زمان ورود
P1	11	0
P2	5	6
P3	4	8
P4	2	10



$$W.T. = \frac{0 + 5 + 10 + 6}{4} = 5.25$$

$$R.T. = \frac{11 + 10 + 14 + 8}{4} = 10.75$$

○ در لحظه ۰:

• فقط فرآیند P1

○ در لحظه ۱۱:

• اولویت P2: ۲

• اولویت P3: ۱.۷۵

• اولویت P4: ۱.۵

○ در لحظه ۱۶:

• اولویت P3: ۳

• اولویت P4: ۴

○ در لحظه ۱۸:

• فقط P3



## الگوریتم LPT (Longest Processing Time)

✓ انتخاب طولانی ترین کار

○ برخلاف SJF

✓ پیاده سازی به صورت انحصاری (معمولا)

✓ احتمال قحطی زدگی برای کارهای کوچک.

✓ حداکثر شدن میانگین زمان پاسخ و زمان انتظار!!



## الگوریتم Lottery

- ✓ توزیع تعدادی بلیط بین فرآیندها.
  - بر اساس اولویت می توان تعداد بلیطهای هر فرآیند را تعیین کرد.
- ✓ اعلام تصادفی یکی از بلیط ها به عنوان برنده در ابتدای هر برش زمانی
  - شانس بالاتر فرآیندهایی که بلیطهای بیشتری دارند.
- ✓ از نوع غیر انحصاری (قابل پس گیری)
- ✓ عدم وجود مشکل قحطی زدگی
- ✓ امکان مبادله بلیطها بین فرآیندها
  - مثلا فرآیندی منتظر I/O است و بلیط خود را اهدا می کند!



## الگوریتم FSS

✓ تلاش برای رعایت عدالت - Fair Share Scheduling

✓ در نظر گرفتن مالک فرآیند برای سهم بندی.

○ ایده های پیشین مالک فرآیندها را در نظر نمی گرفتند.

✓ انگیزه: دو کاربر A و B که کاربر A جمعا ۸ فرآیند دارد و کاربر B

فقط یک فرآیند دارد. آیا باید مالک فرآیندها را در نظر بگیریم؟؟؟

✓ مثال: کاربر A چهار فرآیند  $a_1, a_2, a_3, a_4$  و کاربر B فقط یک فرآیند  $b$  را دارد.

○ اگر سهم هر کاربر ۵۰٪ پردازنده باشد:  $a_1, b, a_2, b, a_1, b, a_2, b, a_1, b, a_2$

○ اگر سهم کاربر A و B به ترتیب ۷۵٪ و ۲۵٪ باشد:

$a_1, a_2, a_3, b, a_4, a_1, a_2, b, a_3, a_4, a_1, b, a_2, a_3, a_4, b, \dots$



## الگوریتم MLQ

- ✓ دسته بندی فرآیندها در چند صف
- ✓ دادن اولویت خاص به هر صف
- ✓ استفاده از یک الگوریتم خاص در هر صف
- ✓ تخصیص پردازنده بین صفها:
  - انحصاری
    - ابتدا باید صف با اولویت بالاتر تکمیل شود.
  - غیر انحصاری
    - شبیه به R.R. مثلا ۵۰٪ سهم صف یک، ۳۰٪ صف دو و ۲۰٪ صف آخر.



## الگوریتم MLFQ

✓ اطلاع ایده MLQ با امکان جابجایی بین صف ها

○ به عنوان مثال یک فرآیند در صف های پایین با افزایش زمان انتظار، به صف های بالاتر منتقل می شود. یا بلعکس.

✓ مثال: می توان سیستمی با چهار صف را در نظر گرفت. در صف اول از RR با برش زمانی ۴ میلی ثانیه و در صف دوم از RR با برش زمانی ۸ میلی ثانیه و در صف سوم از RR با برش زمانی ۱۶ میلی ثانیه استفاده کرد. در صف انتهایی نیز از سیستم FCFS استفاده کرد. حال فرآیندها ابتدا به صف یک وارد می شود، اگر در این صف به پایان نرسیدند، به صف دوم منتقل می شوند و اگر در این صف نیز تکمیل نشدند به صف سوم و چهارم وارد می شوند.





## Thread

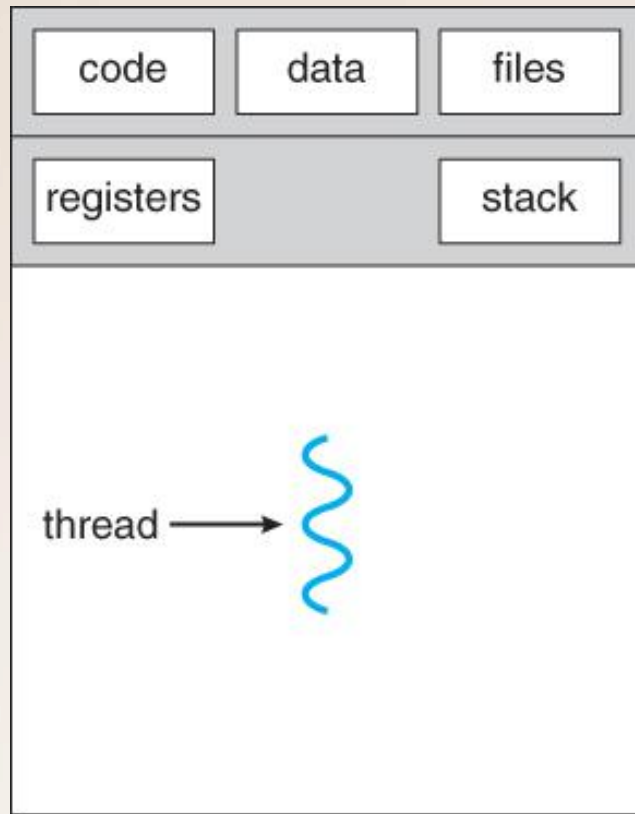
✓ نکات مربوط به نخ

- اجرای مسیرهای مختلف در یک فضای حافظه یکسان از یک فرآیند
- نیاز به حمایت سیستم عامل جهت وجود چند نخ در یک فرآیند
- دسترسی همه نخ ها به فضای حافظه یکسان، متغیرهای سراسری، فایل های باز و ...
- اختصاص یک شمارنده برنامه (P.C.) و ثبات های جداگانه به هر نخ
- نام های مصطلح: فرآیند سبک وزن، نخ، سرنخ، ریسمان، بند، ریسه، رشته، رگه، ریز پردازش.
- مثال مناسب: مرورگر وب و لود کردن تصاویر مختلف به صورت همزمان.

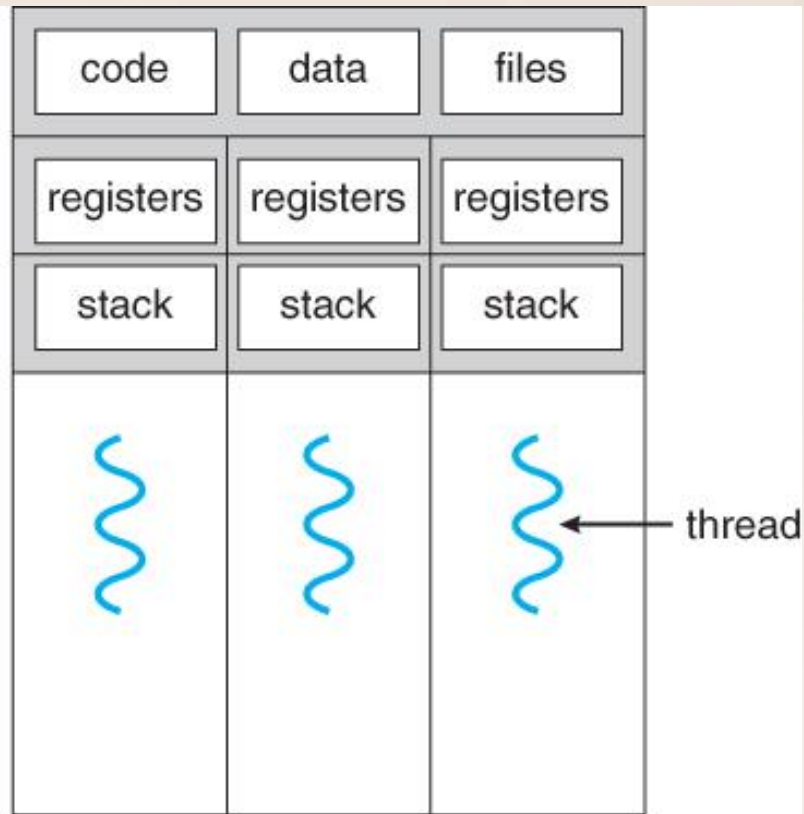


# Thread

✓ نکات مربوط به نخ



single-threaded process



multithreaded process



## Thread

### ✓ مدیریت نخ ها

#### ○ در سطح کاربر

- ایجاد، انتظار، اتمام، زمانبندی نخ ها در فضای کاربر (بدون اطلاع هسته) در داخل یک فرآیند صورت می گیرد.
- هسته فقط فرآیند را می شناسد.
- به عنوان مثال: اگر یک نخ یک فراخوان سیستمی مسدود کننده را اجرا کند، سیستم عامل کل آن فرآیند (شامل همه نخ ها) را مسدود می کند.

#### ○ در سطح هسته

- کلیه اعمال مدیریت و زمانبندی نخ ها توسط سیستم عامل انجام می شود.
- چند پردازنده به طور همزمان می توانند چند نخ یک فرآیند را اجرا کنند.
- اگر یک نخ مسدود شود، سایر نخ ها میتوانند اجرا شوند.



## Thread

### ✓ زمانبندی نخ ها در سطح کاربر

○ مثال: سه فرآیند A و B و C در لحظه صفر آماده زمانبندی در یک سیستم توسط الگوریتم RR با برش زمانی 1.5 واحد زمانی هستند. هر فرآیند شامل نخ هایی به شرح زیر است:

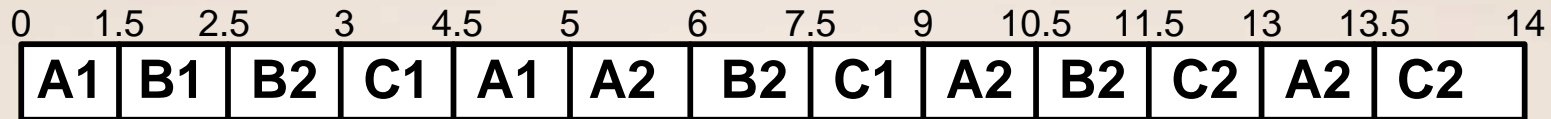
- فرآیند A: نخ A1 با زمان اجرای ۲ و نخ A2 با زمان اجرای ۳
- فرآیند B: نخ B1 با زمان اجرای ۱ و نخ B2 با زمان اجرای ۳
- فرآیند C: نخ C1 با زمان اجرای ۳ و نخ C2 با زمان اجرای ۲

اگر زمانبندی نخ های درون هر فرآیند FCFS باشد و نخ اول هر فرآیند در لحظه آغاز اجرای آن فرآیند و نخ دوم آن فرآیند پس از 0.5 واحد زمانی از لحظه آغاز به کار فرآیند به سیستم وارد شوند، متوسط زمان پاسخ و متوسط زمان انتظار همه نخ های سیستم را محاسبه کنید.



# Thread

○ ادامه مثال قبل



نخ	CBT	زمان ورود	زمان اتمام	زمان پاسخ	زمان انتظار
A1	2	0	5	5	3
A2	3	0.5	13.5	13	10
B1	1	1.5	2.5	1	0
B2	3	2	11.5	9.5	6.5
C1	3	3	9	6	3
C2	2	3.5	14	10.5	8.5



## زمانبندی سیستم های چند پردازنده ای

✓ هدف: معمولاً کاهش طول زمانبندی (زمان مورد نیاز برای اجرای کارها)

○ زمان بندی قابل پس گیری برای کارهای مستقل

• مانند الگوریتم Bin Packing

○ زمان بندی غیرقابل پس گیری برای کارهای مستقل

• مانند الگوریتم LPT



## زمانبندی سیستم های چند پردازنده ای

✓ چند مثال:

○ سه پردازنده داریم، با دو فرآیند که در لحظه صفر وارد شده اند. CBT هر دو فرآیند مساوی و برابر با ۷.۵ ثانیه است. با این سه پردازنده، بهترین حالت زمانبندی کدام است؟

○ سه پردازنده داریم و ۵ فرآیند که در لحظه صفر وارد شده اند. CBT آنها به ترتیب برابر با ۱۰، ۶، ۳، ۴ و ۷ است. بهترین حالت زمانبندی کدام است؟



## زمانبندی سیستم های چند پردازنده ای

### ○ الگوریتم Bin Packing

- توزیع کارها بین پردازنده ها به گونه ای که همیشه پردازنده ها مشغول باشند.
- به حداقل ممکن رساندن طول زمانبندی.
- فرض: دستورات یک فرآیند باید پشت سر هم اجرا شوند.
- ابتدا باید کمترین طول زمانبندی محاسبه شود:

$$W_{opt} = MAX \left\{ \frac{1}{M} \sum T_i, MAX (T_i) \right\}$$

که  $M$  تعداد پردازنده ها و  $T_i$  زمان اجرای کار  $i$ ام

- انتخاب کارها و انتساب آنها به پردازنده ها (ترتیب انتخاب مهم نیست)
- معمولاً آخرین کاری که برای یک پردازنده انتخاب می شود، در محدوده  $W_{opt}$  جا نمیگیرد، باقیمانده کار به پردازنده بعدی سپرده می شود.





## زمانبندی سیستم های چند پردازنده ای

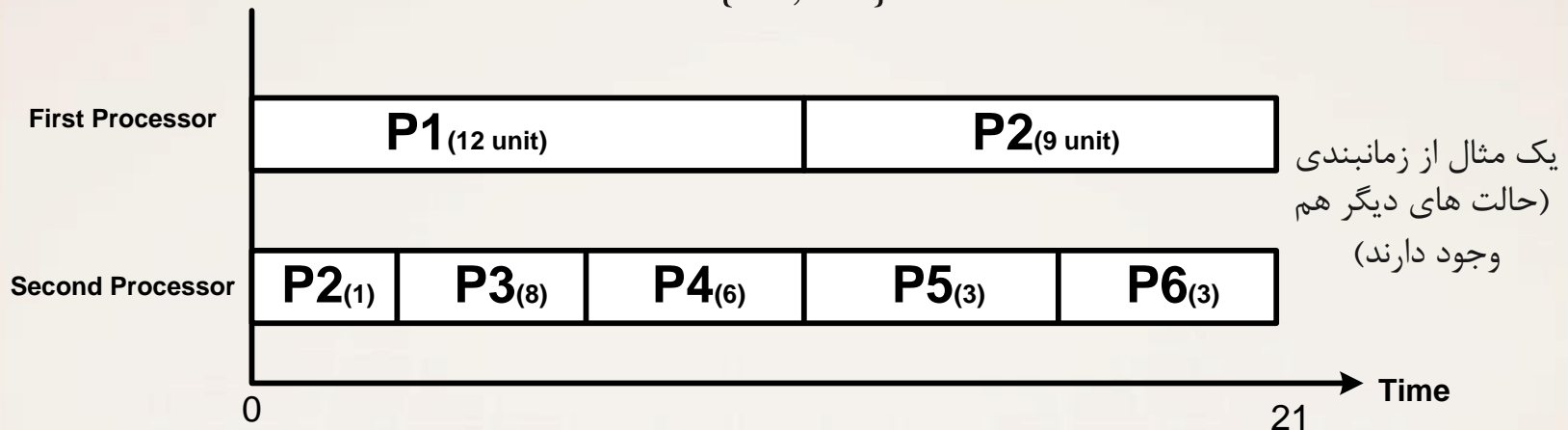
✓ مثال: شش فرآیند زیر را با استفاده از الگوریتم Bin Packing برای حالت های دو پردازنده، سه پردازنده و چهار پردازنده، زمانبندی کنید.

$P1=12, P2=10, P3=8, P4=6, P5=3, P6=3$

حالت اول: دو پردازنده

$$W_{opt} = \text{MAX} \left\{ \frac{1}{2} (12 + 10 + 8 + 6 + 3 + 3), \text{MAX}\{12, 10, 8, 6, 3, 3\} \right\}$$

$$= \text{MAX}\{21, 12\} = 21$$



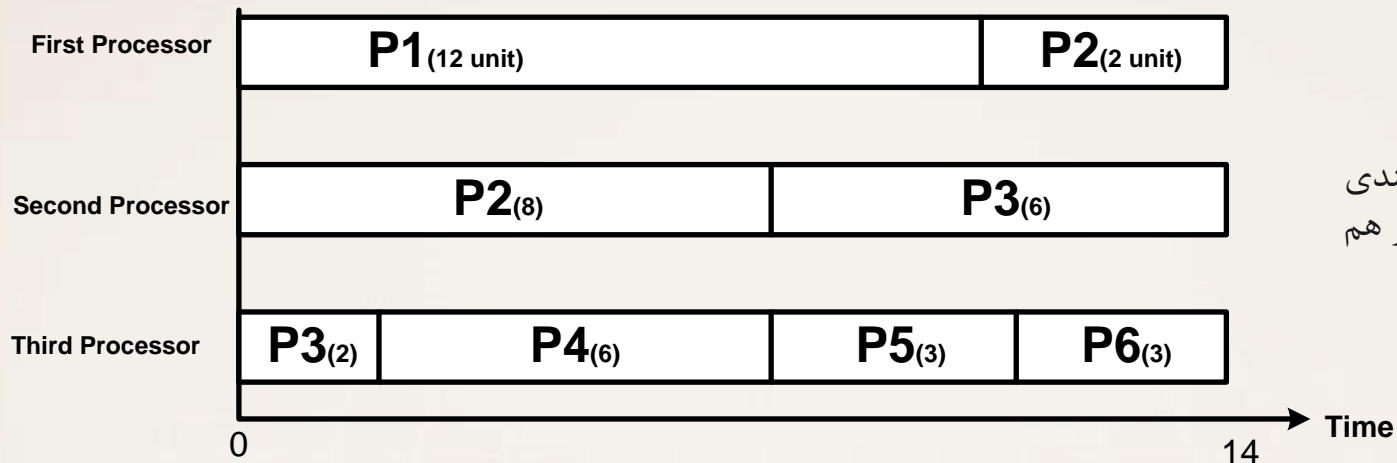


## زمانبندی سیستم های چند پردازنده ای

حالت دوم: سه پردازنده

$$W_{opt} = \text{MAX} \left\{ \frac{1}{3} (12 + 10 + 8 + 6 + 3 + 3), \text{MAX}\{12, 10, 8, 6, 3, 3\} \right\}$$

$$= \text{MAX}\{14, 12\} = 14$$



یک مثال از زمانبندی  
(حالت های دیگر هم  
وجود دارند)

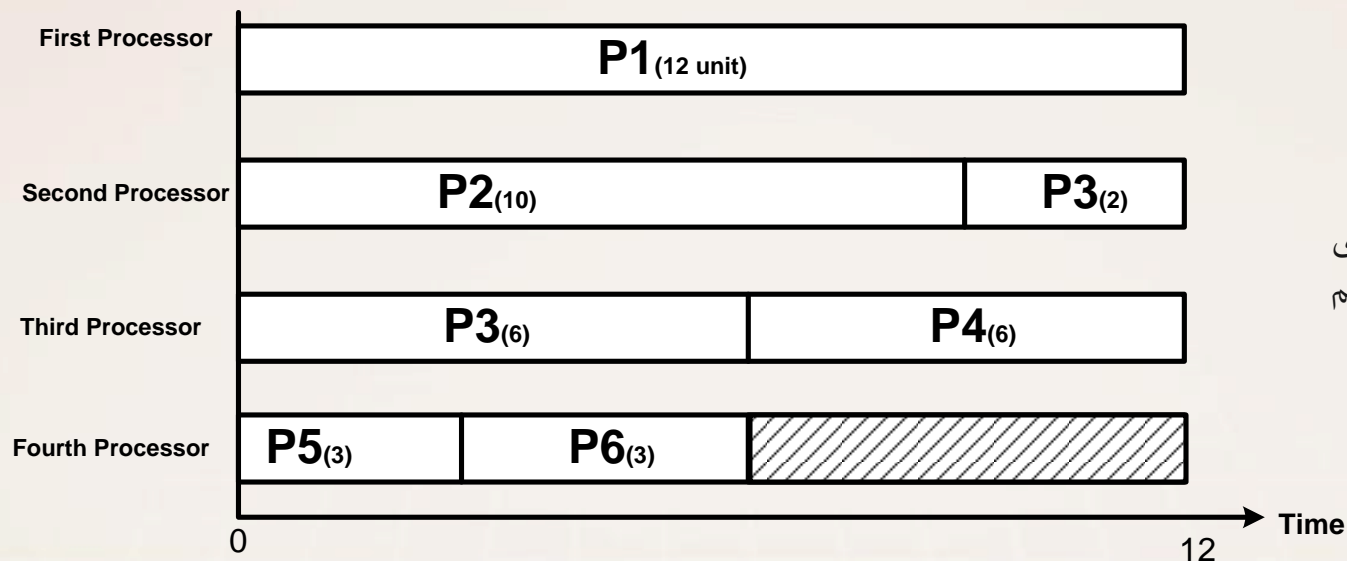


## زمانبندی سیستم های چند پردازنده ای

حالت سوم: چهار پردازنده

$$W_{opt} = \text{MAX} \left\{ \frac{1}{4} (12 + 10 + 8 + 6 + 3 + 3), \text{MAX}\{12, 10, 8, 6, 3, 3\} \right\}$$

$$= \text{MAX}\{10.5, 12\} = 12$$



یک مثال از زمانبندی  
(حالت های دیگر هم  
وجود دارند)



## زمانبندی سیستم های چند پردازنده ای

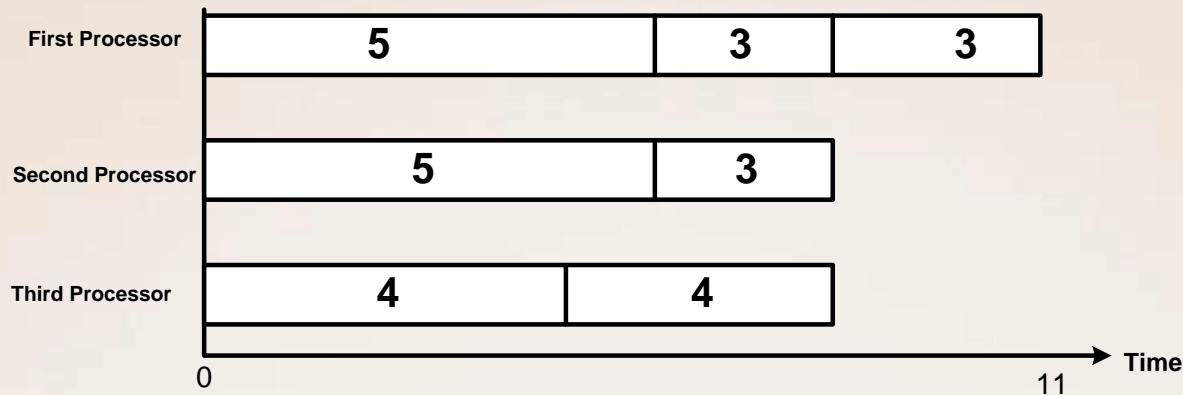
✓ الگوریتم LPT (برای زمانبندی غیرقابل پس گیری)

- در سیستم های انحصاری رسیدن به یک زمانبندی با طول بهینه عملاً غیر ممکن است.
- الگوریتم LPT در صورت به کار گیری در سیستم های چند پردازنده ای معمولاً منجر به زمانبندی با طول معقول (نه بهینه) می شود.
- روال کار: هر گاه یک پردازنده آزاد است، از بین کارهای باقیمانده، طولانی ترین کار را انتخاب می کند.
- الگوریتم LPT از نظر طول زمانبندی معمولاً خوب عمل می کند. اما زمان پاسخ را افزایش می دهد.

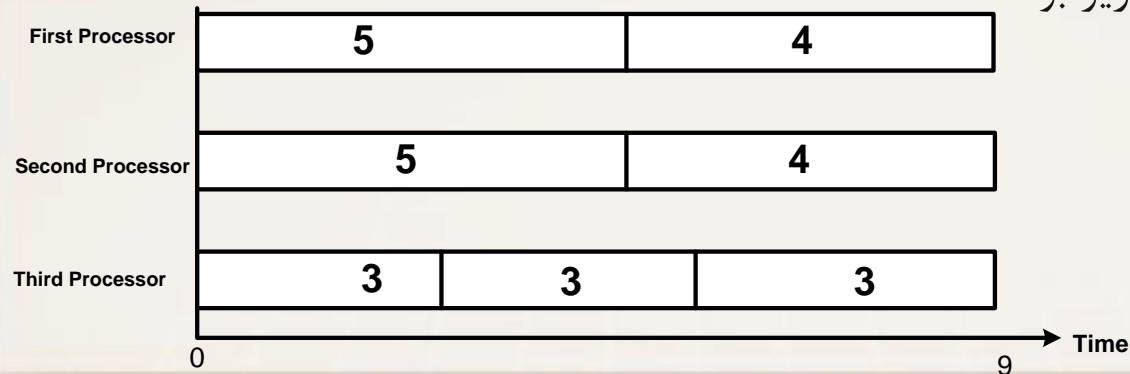


## زمانبندی سیستم های چند پردازنده ای

✓ مثال: سیستمی با ۳ پردازنده را در نظر بگیرید. هفت فرآیند با طول های  $\{5, 5, 4, 4, 3, 3, 3\}$  را با استفاده از LPT زمانبندی کنید.



✓ اما حالت بهینه به صورت زیر بود:





## زمانبندی سیستم های چند پردازنده ای

✓ نقص عمده LPT: افزایش میانگین زمان پاسخ فرآیندها

○ یک ایده بهبود: زمانبندی با همان ایده LPT و سپس معکوس کردن کارهای هر پردازنده! (معروف به ایده Reverse Processing Time)

✓ جهت کاهش میانگین زمان پاسخ، می توان از الگوریتم SPT استفاده کرد.

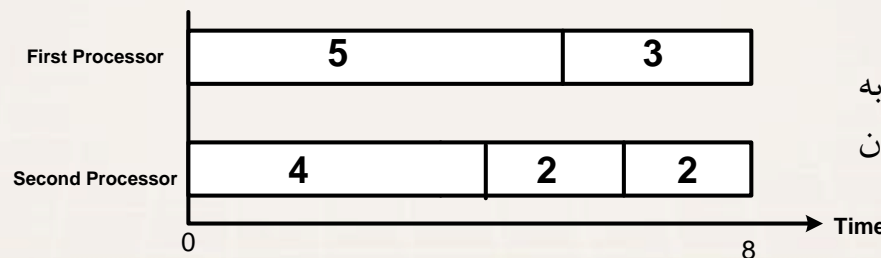
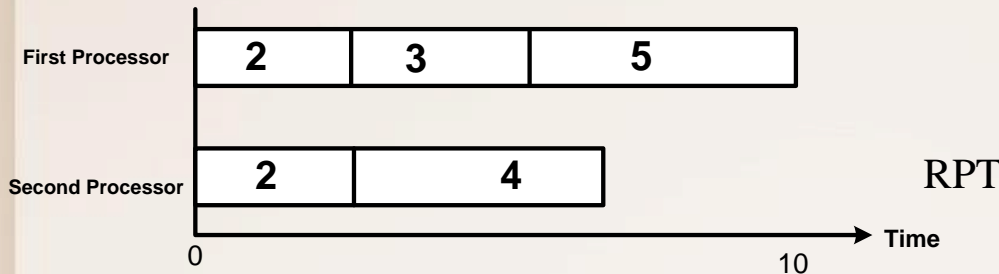
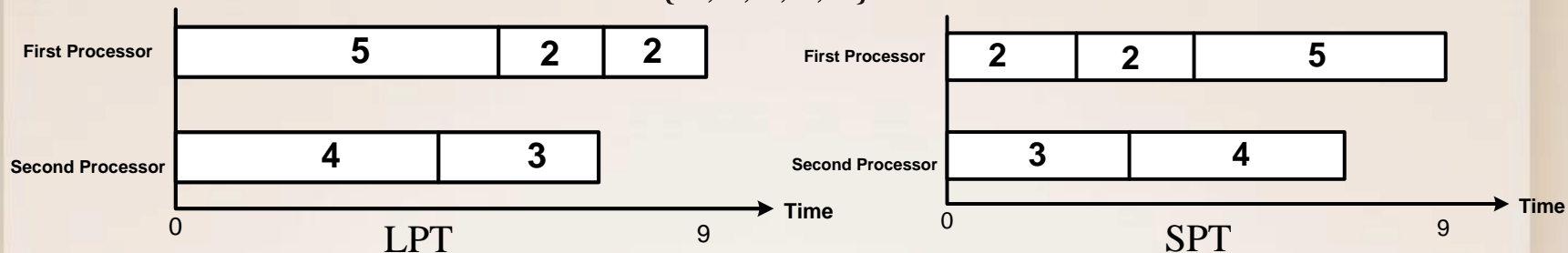
○ افزایش طول زمانبندی به بدترین حالت ممکن!



## زمانبندی سیستم های چند پردازنده ای

مثال: پنج فرآیند زیر را برای دو پردازنده با الگوریتم های LPT، RPT و SPT زمانبندی کنید. ✓

{5,4,3,2,2}



اما در واقع حالت بهینه به صورت روبرو است که کشف آن به راحتی صورت نمی گیرد!



## سوال ها

✓ سوال ۱: درباره فرمول Little پیرامون رابطه پایداری سیستم با تعداد فرآیندها و زمان سرویس فرآیندهای سیستم تحقیق کنید.  
 ○ راهنمایی: در مرجع [1] بخش 2-8-5 در این مورد صحبت شده است.

✓ سوال ۲: در مورد پدیده وارونگی اولویت تحقیق کنید.  
 ○ منبع [1]، بخش 2-8-6.

✓ سوال ۳: درباره شرط پاسخ گویی پردازنده به وقایع در سیستم های بلادرنگ تحقیق کنید.  
 ○ راهنمایی: در مرجع [2] بخش 4-4-2 در این مورد بحث شده است.





## سوال ها

✓ سوال ۴: درباره اهداف روش های زمانبندی در سیستم های مختلف تحقیق کنید.

○ راهنمایی: مرجع [4] شکل 2-23.

✓ سوال ۵: درباره فراخوان سیستمی fork در یونیکس تحقیق کنید.

○ منبع: آزاد

✓ سوال ۶: درباره الگوریتم زمانبندی CPU در Linux و Windows 2000 تحقیق کنید.

○ راهنمایی: مرجع [3] بخش های 10-3 و 10-6



## سوال ها

✓ سوال ۷: فرض کنید از یک الگوریتم زمانبندی ابداعی استفاده می کنیم به این صورت که فرآیندهایی برای اجرا انتخاب می شوند که در گذشته اخیر، کمترین استفاده را از زمان پردازنده داشته اند، در مورد رفتار این الگوریتم در قبال فرآیندهای با عطش I/O و فرآیندهای با عطش پردازنده بحث کنید.

✓ سوال ۸: درباره انواع زمانبندی سیستم های بلادرنگ تحقیق کنید.  
○ راهنمایی: مرجع [1] بخش 5-6 و زیر بخش های مربوطه. چندین الگوریتم ذکر شده را بررسی کنید.



## سوال ها

✓ سوال ۹: در مورد چند وظیفه ای در سیستم عامل های مبتنی بر موبایل، مانند iOS و اندروید تحقیق کنید.

○ مرجع [1]، صفحه ۱۱۵.

✓ سوال ۱۰: مفهوم Cascading Termination چیست؟

○ مرجع [1]، بخش 2-3-3.

✓ سوال ۱۱: در مورد Pop-up Thread ها تحقیق کنید و مزایا و معایب اجرای آن ها در فضای Kernel را توضیح دهید.

○ مرجع [۲] – بخش ۲.۲.۸



## سوال ها

✓ سوال ۱۲: تعریف و علت استفاده Implicit Threading را بیان کنید و روش های ارائه شده برای programming که از آن استفاده می کنند را به طور مختصر شرح دهید.

○ مرجع [۱] - بخش ۴.۵

✓ سوال ۱۳: هنگام ایجاد سرخ ها، تفاوت بین سرخ های سنکرون و آسنکرون چیست؟

○ مرجع [1]، بخش 4-4.

✓ سوال ۱۴: نخ ها در ویندوز چه شباهت ها و تفاوت های رفتاری با Pthread دارند؟

○ مرجع [1]، بخش های 4-4-1 و 4-4-2.



## سوال ها

✓ سوال ۱۵: استخر نخ ها چیست و برای حل چه مشکلاتی از آن استفاده می شود؟  
○ مرجع [1]، بخش 1-5-4.

✓ سوال ۱۶: استاندارد IEEE 1003.1c در مورد نخ ها را به اختصار بررسی کنید.  
○ مرجع [2]، بخش 3-2-2.

✓ سوال ۱۷: حل تشریحی تست های انتهای اسلاید را به طور کامل و دقیق انجام دهید.

- در صورت نیاز، حل تشریحی را به صورت خوانا و خوش خط روی کاغذ نوشته و تصویر آن را ضمیمه کنید.



## پروژه فصل دو

✓ برای این فصل دو پروژه برنامه نویسی در نظر گرفته شده است که جزئیات، نحوه پیاده سازی و ارائه آنها در کلاس های تدریسار ارائه می گردد.



## تست های کنکور ارشد

### ✓ ارشد، دولتی، ۷۶

○ در صورتی که چهار پردازش A و B و C و D به همین ترتیب در لیست پردازش های آماده اجرای یک زمانبند قرار داشته باشند، زمان اجرای تخمینی آنها به ترتیب برابر ۴۰ و ۲۰ و ۵۰ و ۳۰ میلی ثانیه باشد و زمان هر Context Switch بین پردازش ها برابر ۵ میلی ثانیه باشد و از روش RR با برش زمانی ۲۰ میلی ثانیه استفاده شود، متوسط زمان پاسخگویی و متوسط زمان انتظار پردازش ها چقدر است؟

(د) ۱۳۵ و ۱۰۵

(ج) ۱۲۸.۷۵ و ۱۱۰

(ب) ۱۰۲.۵ و ۶۷.۵

(الف) ۱۲۵ و ۹۰

جواب: الف

### ✓ ارشد، آزاد، ۸۱

○ چهار فرآیند مطابق جدول زیر در سیستم وجود دارند، اگر از روش RR با برش زمانی یک میلی ثانیه استفاده شود و از سربار تعویض متن صرف نظر شود، میانگین زمان انتظار فرآیند ها چقدر است؟

	P1	P2	P3	P4	
زمان ورود(دقیقه)	0	1	3	9	
زمان اجرا (دقیقه)	3	5	2	2	

(الف) ۳ دقیقه

(ب) ۶ دقیقه

(ج) ۳ دقیقه و ۳۰ ثانیه

(د) ۶ دقیقه و ۳۰ ثانیه

• جواب: الف



## تست های کنکور ارشد

### ✓ ارشد، دولتی، ۸۱

○ دو فرآیند P1 و P2 با مشخصات اجرای زیر در سیستم موجودند، اطلاعات هر سطر منبع مورد نیاز برای هر پراسس و زمان مورد نیاز را مشخص می کند. مثلا net3 در سطر چهارم بیانگر این است که پراسس دوم کارت شبکه را به مدت ۳ ثانیه نیاز دارد. اگر پراسس P2 دقیقاً ۲ ثانیه بعد از P1 به سیستم رسیده باشد، و سیستم سیاست SJF با خاصیت Preemption را برای برنامه ریزی پراسس ها اعمال کند، زمان اجرای دو پراسس مذکور و زمان هدر رفتگی وقت CPU بر حسب ثانیه چقدر است؟

P1	P2
CPU3	CPU4
Net4	Disk3
CPU2	CPU3
Disk3	Net3
CPU5	CPU3
DISK2	Net3
CPU2	CPU3

- الف) کل زمان ۲۴ و هدر رفتگی ۰
- ب) کل زمان ۲۵ و هدر رفتگی ۱
- ج) کل زمان ۲۷ و هدر رفتگی ۲
- د) کل زمان ۲۸ و هدر رفتگی ۳

- جواب: ۳
- راهنمایی: یک گانت چارت ابداعی همزمان برای هر دو فرآیند رسم کنید.
- نکته: دستگاههای ورودی خروجی در آن واحد در اختیار یک کار هستند.





## تست های کنکور ارشد

### ✓ ارشد، دولتی، ۸۰

○ یک سیستم تک پردازنده ای با صف بازخورد چند سطحی (MLFQ) را در نظر بگیرید. به صف اول تکه زمانی معادل ۸ میکرو ثانیه و به صف دوم تکه زمانی معادل ۱۶ میکرو ثانیه داده شده است و صف سوم به صورت FCFS زمانبندی شده است. فرض کنید شش کار همگی در زمان صفر به سیستم وارد شوند و زمان اجرای آنها به ترتیب ۴ و ۷ و ۱۲ و ۲۰ و ۲۵ و ۳۰ میکرو ثانیه باشد. میانگین زمان پاسخ و میانگین زمان انتظار را بیابید.

الف) ۱۶.۳۳ و ۳۵.۵ (ب) ۵۱.۸۳ و ۳۵.۵ (ج) ۴۸.۵ و ۱۶.۳۳ (د) ۵۱.۸۳ و ۲۴.۷۵

• جواب: ب

• راهنمایی: در این حالت فرض شده است که پردازنده ابتدا به صف اول یک دور سرویس می دهد، فرآیندهایی که اتمام نمی یابند به صف بعدی وارد می شوند. بعد پردازنده سراغ صف دوم می رود و بعد سراغ صف سوم. یعنی فرآیندها در صف اول و دوم فقط یک دور سرویس میگیرند و در نهایت در صف سوم ماندگار می شوند.

### ✓ ارشد، آزاد ۸۳

○ یک سیستم تک پردازنده ای از الگوریتم SRT استفاده می کند. چهار فرآیند با زمان اجرای تخمینی ۶ و ۳ و ۲ و ۳ میلی ثانیه وارد سیستم می شوند. اگر زمان تعویض متن ناچیز باشد و تمامی فرآیندها فقط کار پردازشی داشته باشند، نگاه میانگین زمان انتظار فرآیندها چند میلی ثانیه است؟

الف) ۱.۵ (ب) ۰.۲۵ (ج) ۲.۵ (د) ۲.۲۵

• جواب: د



## تست های کنکور ارشد

### ✓ ارشد، آزاد، ۸۴

○ یک سیستم تک پردازنده ای از الگوریتم زمانبندی HRRN استفاده می کند. در صورتی که فرآیندهایی با زمان اجرای ۶ و ۷ و ۳ و ۳ و ۲ به ترتیب در زمانهای ۰ و ۱ و ۳ و ۵ و ۷ وارد سیستم شوند. کدام فرآیند در انتها اجرا می شود؟

الف) فرآیندی که در زمان ۱ وارد شده.

ب) فرآیندی که در زمان ۳ وارد شده.

ج) فرآیندی که در زمان ۵ وارد شده.

د) فرآیندی که در زمان ۷ وارد شده.

• جواب) الف

### ✓ ارشد، آزاد، ۸۵

○ سیستمی از الگوریتم SRT استفاده می کند. اگر زمان تعویض فرآیندها ۱ میلی ثانیه باشد، متوسط زمان انتظار برای اجرای فرآیندها چقدر است؟

	P0	P1	P2	P3
زمان ورود	0	2	3	8
زمان اجرا	6	4	2	1

الف) ۵.۲۵

ب) ۴.۷۵

ج) ۵

د) ۴.۵

• جواب) الف



## تست های کنکور ارشد

### ✓ ارشد، آزاد، ۸۶

○ در یک سیستم هفت پراسس با استفاده از الگوریتم زمانبندی RR اجرا می شوند. اگر زمان اجرای آنها به ترتیب ۵ و ۴ و ۳ و ۴ و ۲ و ۶ میلی ثانیه باشد، برای افزایش بهره وری CPU مقدار کوانتوم (برش) زمانی چقدر باشد؟

الف) ۶      ب) ۵      ج) ۴      د) ۷

### ✓ ارشد، آزاد، ۸۷

○ در سیستمی به طور متوسط در هر ۵۰ ثانیه، ۱۰ فرآیند وارد می شوند. به طوری که میانگین زمان سرویس هر فرآیند، ۲ ثانیه است. اگر زمانبندی از نوع RR با کوانتوم زمانی ۱۰۰ میلی ثانیه باشد و زمان تعویض فرآیند ناچیز باشد، میانگین بار سیستم کدام است؟

الف) ۶۰ درصد      ب) ۴۰ درصد      ج) ۸۰ درصد      د) اطلاعات داده شده کافی نیست

• راهنمایی: بار سیستم = بهره وری پردازنده



## تست های کنکور ارشد

✓ ارشد، آزاد، ۹۰

○ چهار فرآیند در یک لحظه در صف آماده قرار دارند. زمان تخمین اجرای آنها به ترتیب از P1 تا P4 برابر X و ۷ و ۵ و ۸ می باشد. چنانچه حداقل میانگین زمان پاسخ فرآیندها برابر ۱۳.۵ باشد مقدار X را بیابید. (فرض بر این است که  $X < 5$  است)

۴ (د)

۴.۵ (ج)

۴.۲۵ (ب)

۴.۷۵ (الف)



## منابع

- [1]. A. Silberschatz, P. B. Galvin and G. Gagne, **“Operating System Concepts,”** 10<sup>th</sup> ed., John Wiley Inc., 2018. ([Download link](#))
- [2] A. S. Tanenbaum and H. Bos, **“Modern Operating Systems,”** 4<sup>rd</sup> ed., Pearson, 2015. ([Download Link](#))
- [3] W. Stallings, **“Operating Systems,”** 9<sup>th</sup> ed., Pearson, 2018. ([Download Link](#))
- [4] A. S. Tanenbaum, A. S. Woodhull, **“Operating Systems Design and Implementation,”** 3<sup>rd</sup> ed., Pearson, 2006. ([Download Link](#))
- [5] نستوه طاهری جوان و محسن طورانی، "اصول و مفاهیم سیستم عامل"، انتشارات موسسه آموزش عالی پارسه، ۱۳۸۶. (لینک داندود)

برای داندود کتاب ها، اسلایدها و نمونه پروژه های درسی به سایت [www.nastoooh.com](http://www.nastoooh.com) بخش دانشجویان مراجعه کنید.



پایان